Amendments to the Claims

Listing of Claims:

1.   (Currently amended) A method for restoring a persistently stored objects of an object oriented environment established on a computer system having a volatile memory and a persistent storage, the method comprising ~~the~~ steps of:

retrieving from said persistent storage a first list of ~~comprising~~ first references to segments stored in said persistent storage;

retrieving all segments referenced ~~by~~ in said the first ~~references~~ list and storing ~~them~~ the segments in said volatile memory;

saving in said first list ~~the~~ a difference between an old memory address, at which the segment used to reside in the volatile memory, and a new memory address at which said segment is currently stored;

retrieving from said persistent storage a second list of ~~comprising~~ second references to blocks, wherein the blocks are portions of segments and whereby ~~one or more~~ at least one of said blocks contains an object description;

determining, from the second list, which segment contains ~~said~~ the block ~~referenced by a particular element of said second list~~ where the object description of the persistently stored object is located;

having determined the segment in which the object description of the persistently stored object resides, retrieving the objection description;

creating a ~~new~~ first object in said volatile memory using said object description retrieved from said segment; and

saving a new address of said ~~new~~ first object in said second list in volatile memory;

initializing said ~~new~~ first object with values taken from said object description; ~~and~~

determining ~~said new addresses of objects referenced by the object and setting said~~ ~~new~~ the address ~~as the reference in said new~~ of a second object description of a second object referenced in said first object; and

setting the address of said second object description as a reference in the object

2

<u>description of the first object</u>.

2. (Previously presented) The method according to claim 1, whereby said first list and/or said second list are ordered lists.

3. (Currently amended) The method according to claim 1 ~~or 2~~, whereby said first list and/or said second list are organized as a B-tree.

4. (Currently amended) The method according to claim ~~1~~ 2, whereby ~~the~~ elements of said first ordered list are indexed by said first references.

5. (Currently amended) The method according to claim 1, whereby each of said first references corresponds to the old memory address at which the ~~respective~~ segment used to reside in the volatile memory.

6. (Currently amended) The method according to claim ~~1~~ 2, whereby ~~the~~ elements of said second ordered list are indexed by said second references.

7. (Currently amended) The method according to claim 1, whereby each of said second references corresponds to ~~the~~ an old memory address at which the ~~respective~~ block used to reside in said volatile memory.

8. (Currently amended) The method according to claim 1, whereby said <u>first and second</u> object descriptions <u>is</u> <u>are</u> formed by a collection of values owned by an object for ~~the~~ variables belonging to its class.

9. (Currently amended) The method according to claim 1, whereby for each value in said <u>first</u> <u>and second</u> object description<u>s</u> of variables having a variable length, the method further comprising ~~the~~ steps of:

allocating a number of blocks<u>, such that</u> ~~that allows to keep the~~ <u>an</u> actual value of the variable having <u>a</u> variable length <u>can be kept variable</u>;

creating a linked list of said number of blocks;

saving said value into said number of blocks; and

storing a reference to the head of the linked list in said <u>first and second</u> object description<u>s</u>.

10. (Currently amended) The method according to claim ~~1~~<u>2</u>, whereby determining the segment that contains said block referenced by a particular element of said second list comprises the step of searching in said first ordered list ~~(segment map)~~ for the segment that contains said portion of said segment ~~(block)~~ referenced by said element.

11. (Currently amended) The method according to claim 1, whereby determining the segment that contains said block referenced by a particular element of said second list further comprises the step of calculating the new address by adding the reference to said block<u>,</u> ~~(that corresponds to the old memory address,)~~ and said difference between said old memory address and said new memory address.

12. (Currently amended) The method according to claim 1, whereby determining ~~the~~ new addresses of objects referenced by ~~the~~ <u>a</u> newly created object comprises the step of searching in said second list ~~(object map)~~ for the element ~~said~~ <u>that</u> contains the new address of the referenced object, that is referenced by the old address of the respective object description.

13. (Previously presented) The method according to claim 1, whereby for all references to heads of linked lists the method further comprising the steps of:

reading all blocks of said linked list;

allocating memory to store the value of the variable retrieved from the linked list; and

storing the value in said allocated memory.

14. (Currently Amended) A method for persistently storing objects of an object-oriented environment established on a computer system having a volatile memory and a persistent storage, the method comprising steps of:

allocating a plurality of segments in said volatile memory;

creating a first list ~~comprising~~ of first references to said segments;

creating a second list ~~comprising~~ of second references to blocks, wherein the blocks are portions of said segments;

allocating a block of one of said segments~~;~~:

creating a first object description for a first object, wherein the first object description comprises ~~by saving~~ values ~~owned by the first object~~ of ~~the~~ variables belonging to ~~its~~ the first object's class;

storing the first object description into said allocated block; ~~and~~

saving a new address of said first object description in the second list, wherein the new address is a memory address of said allocated block in the volatile memory where the first object description is stored;

adding ~~a new element to said second list containing~~ a ~~particular~~ reference to said first object description to the second list;

determining the address of a second object description of a second object referenced in said first object description;

setting the address of said ~~respective~~ second object description as ~~the~~ a reference in the created first object description;

storing said second list on said persistent storage;

storing the plurality of segments referenced by said first list on said persistent storage; and

storing said first list on said persistent storage.

15. (Previously presented) The method according to claim 14, whereby said first list and/or said second list are ordered lists.

6

16. (Currently amended) The method according to claim 14, whereby said first list ~~and/~~or said second list are organized as a B-tree.

17. (Currently amended) The method according to claim ~~14~~ 16, whereby ~~the~~ elements of said first ordered list are indexed by said first references.

18. (Currently amended) The method according to claim 14, whereby each of said first references corresponds to the current memory address at which the ~~respective~~ segment resides in the volatile memory.

19. (Currently amended) The method according to claim ~~14~~ 15, whereby ~~the~~ elements of said second ordered list are indexed by said second references.

20. (Currently amended) The method according to claim 14, whereby each of said second references corresponds to the current memory address at which the ~~respective~~ block resides in said volatile memory.

21. (Currently amended) The method according to claim ~~14~~ 19, whereby determining ~~the~~ an address of ~~the~~ an object description of another object referenced in said first object comprises the step of searching in said second ordered list for the element ~~said~~ that contains the address of the object description of the ~~referenced~~ another object.

22. (Currently amended) The method according to claim ~~14~~ 21, whereby determining the address of the object description of another object referenced in said first object further comprises a step of using a reference to the ~~respective~~ another object description provided by each object in said second ordered list.

7

23. (Currently amended) The method according to claim 14, whereby for each value of variables having a variable length the method further comprises steps of:

allocating a number of portions of one of said ~~pieces~~ segments of memory such that ~~allows to keep the~~ an actual value of the variable length can be kept variable;

creating a linked list of said number of portions;

saving value into said number of portions; and

storing a reference to the head of the linked list in said object description.

24. (Currently Amended) A computer program product stored on a computer usable medium, comprising computer readable program instructions for:

allocating a plurality of segments in ~~said~~ volatile memory ~~segments~~;

creating a first list of ~~comprising~~ first references to said segments;

creating a second list of ~~comprising~~ second- references to blocks, wherein the blocks are portions of said segments;

allocating a block of one of said segments,

creating ~~an~~ a first object description of a first object, wherein the first object description comprises ~~by saving~~ values ~~owned by the object~~ of ~~the~~ variables belonging to ~~it's~~ the first object's class;

storing the first object description into said allocated block;

saving a new address of said first object description in the second list, wherein the new address is a memory address of said allocated block in the volatile memory where the first object description is stored;

adding a ~~new element to said second list containing the particular~~ reference to ~~said created~~ the first object description to the second list;

determining the address of ~~the~~ a second object description of a ~~another~~ second object referenced in said first object description;

setting the address of said ~~respective~~ second object description as ~~the~~ a reference in the created first object description;

storing said second list on ~~said~~ persistent storage;

storing the plurality of segments referenced by said first list on said persistent storage; and

storing said first list on said persistent storage.

.